Python

Errors and Exception Handling

@kabirbaidhya

Reflections

What we already know

From the last session.

1. Python basics, conditionals and Loops
2. Functions

Note: If you're not aware of these. Read them at https://github.com/kabirbaidhya/learn-python-django-web

Exceptions

Exceptions

When there are errors in your syntax, your code won't run, that is for sure. But even if your syntax is 100% correct and runs fine, there could be cases when you get errors during the runtime of your program.

These errors that get triggered in the runtime are called **Exceptions**.

For instance

Consider this example, when you're trying to divide two numbers, both received from the user input.

```
a = float(input('First Number: '))
b = float(input('Second Number: '))

result = a / b
```

It would raise a runtime error `ZeroDivisionError` and the program would halt.

Handling Exceptions

Handling Exceptions

We can handle exceptions using `try` block.

Handling exceptions ensures that the program still continues to run regardless of the exceptions.

```python
a = float(input('First Number: '))
b = float(input('Second Number: '))

try:
    result = a / b
except ZeroDivisionError:
    print('Error: Division by Zero')
```

Improved Example

```python
while True:
    a = float(input('First Number: '))
    b = float(input('Second Number: '))

    try:
        result = a / b
        print('Result = {}'.format(result))

    except ZeroDivisionError:
        print('Error: Division by Zero')

    try_again = input('\nTry Again (Y/N)? ')

    # If the user doesn't want to try again exit the loop.
    if try_again.upper() != 'Y':
        break

    print()

# Program will exit finally
print('Good Bye!')
```

The `try` statement

You've already seen how and why we use `try` statement. Let's get into details about how it actually works.

Syntax

```
try:
    STATEMENTS
except SomeException:
    # Code to handle exception
```

How it works?

Firstly the statements inside the `try` block are executed. If any statement causes exceptions the rest of the code in the block are skipped.

If the raised exception is there in the `except` clause, then that particular except block is executed.

In case the raised exception is not in the `except` clause it will propagate to the higher level. If it couldn't find any handlers even after reaching the highest level, the program terminates with that exception.

If no exception occurs inside the `try` block, the `except` blocks are skipped.

Example 1

Consider the following example we did in our previous lesson.

```python
n, sum = 0, 0

while n < 5:
    value = input('Enter Number %s: ' % (n + 1))
    sum = sum + float(value)
    n += 1

print('Sum = %.2f' % sum)
```

This program expects numeric values from the user.

Example 1 - Error

Go and try it with some input like `abx`, `xyz` etc.

You'll get an error like this:

```
Traceback (most recent call last):
  File "units/python/4/example_5.py", line 7, in <module>
    sum = sum + float(value)
ValueError: could not convert string to float: 'abc'
```

Example 1 - Exception Handled

Let's handle this error now.

```
n, sum = 0, 0

while n < 5:
    value = input('Enter Number %s: ' % (n + 1))

    try:
        value = float(value)
        sum = sum + value
        n += 1

    except ValueError:
        print('Invalid Input. Please enter a numeric value.

print('\nSum = %.2f' % sum)
```

Possible Variations

of `try...except` statements

Multiple `except` blocks

There could be any number of `except` clauses following a `try` statement.

```python
try:
    STATEMENTS
except SomeException:
    # Code to handle exception
except SomeOtherException:
    # Code to handle exception
except AndAnotherException:
    # Code to handle exception
```

Multiple exceptions in one `except` clause

A single `except` clause can also accept multiple exceptions as parenthesized tuple.

```
try:
    STATEMENTS
except (RuntimeError, TypeError, NameError):
    # Code to handle exception
```

The exception instance

You can get the instance of the actual error or exception object using the following syntax.

```
try:
    STATEMENTS
except SomeException as e:
    # Do something with this `e`
```

Raising Exceptions

We can use the `raise` keyword to raise exceptions like this:

```
raise Exception('Hey, this was a test exception.')
raise ValueError('Hey, this was another exception.')
```

The only argument required for the `raise` keyword is the exception itself. This could be either an exception instance or exception class(a class that derives from Exception).

Built-in Exceptions

There are various types of exceptions in python. Check the official docs to know about the Built-in Exceptions in python.

Exercises

Exercise 1

Improvements on the program to find the area of circle.

```
- Move the logic to compute the area to a function
- Handle runtime exceptions
- Ability to try again in case of invalid input.
```

Exercise 2

Improvements on the program we did to compute the age of the user by checking his date of birth.

```
- Refactor it using functions.
- Handle runtime exceptions.
- Ability to try again in case of invalid input.
```

Exercise 3

Improvements on the program to parse out the value of `m` and `c` from the equation of line `y = mx + c`

```
- Refactor the logic for parsing the equation to a function
- Take two user inputs: equation of two lines
- Write a function to get the intersection of two lines
- Write a function to get the angle between two lines
- Print angle between two lines and the point of intersecti
- Print if they're parallel or perpendicular to each other
- Handle runtime exceptions.
- Ability to try again in case of invalid input.
```

Exersise 4

Program to ask for a filename and read the contents of the file and print it on the screen. Ensure there are no unhandled exceptions.

Read More?

Links

Want to read more? Go through these links.

1. https://docs.python.org/3/tutorial/errors.html
2. https://wiki.python.org/moin/HandlingExceptions
3. https://docs.python.org/3/library/exceptions.html
4. http://stackoverflow.com/questions/730764/try-except-in-python-how-do-you-properly-ignore-exceptions
5. https://www.tutorialspoint.com/python/python_exceptions.htm

This slide was a part of course

Python, Django & Web Development

github.com/kabirbaidhya/learn-python-django-web

Thank You

@kabirbaidhya

kabirbaidhya@gmail.com

The slides were created using Marp. https://yhatt.github.io/marp/